1. Title of Thesis:

A Refinement Approach to Online Data Mining for Performance Improvement

2. Introduction

The field of data mining and knowledge discovery is emerging as a new, fundamental research area with important applications to science, engineering, medicine, business, and education. Data mining attempts to formulate, analyze and implement basic induction processes that facilitate the extraction of meaningful information and knowledge from unstructured data. Data mining extracts patterns, changes, associations and anomalies from large data sets.

Many fund management firms have invested heavily in information technology to help them manage their financial portfolios. Over the last three decades, increasingly large amounts of historical data have been stored electronically and this volume is expected to continue to grow considerably in the future. Yet despite this wealth of data, many fund managers have been unable to fully capitalize on their value. This is because information that is implicit in the data for the purpose of investment is not easy to discern.

When market-beating strategies are discovered via data mining, there are a number of potential problems in making the leap from a back-tested strategy to successfully investing in future real world conditions. The first problem is determining the probability that the relationships are not random at all market conditions. This is done using large historic market data to represent varying conditions, say 10 years of data, and confirming that the time series patterns have statistically significant predictive power for (1) high probability of profitable trades and (2) high profitable returns for the investment.

In order to do this analysis, several different types of tasks have been identified, corresponding to the objectives of what needs to be analyzed and more importantly, what the intended outcome should describe. These tasks can be categorized as follows.
Exploratory Data Analysis: The goal is here to explore the data without any clear ideas of what is desired. Typical techniques include graphical display methods, projection techniques and summarization methods.
Retrieval by Content: The user has a specific pattern in mind in advance and is looking for similar patterns in the data set. This task is most commonly used for the retrieval of information from large collections of text or image data. The main challenge here is to define similarity and how to find all similar patterns according to this definition. A well-known example is the Google search engine (http://www.google.com) of Brin and Page, which finds web pages that contain information similar to the set of key words given by the user.

Descriptive Modeling: As the name suggests, descriptive models try to describe all of the collected data. Typical descriptions include several statistical models, clusters and dependency models.

3. A brief review of the work already done in the field:

3.1Association Rule Mining

The aim of Association Rule Mining is to find latent associations among data entities in database repositories, a typical example of which is the transaction database maintained by a supermarket. An association rule is an implication of the form $A \Rightarrow B$ which conveys that customers buying set of items A would also with a high probability buy set of items B.

The concept of association rule mining was first introduced in [AS93]. Typically the problem is decomposed into two phases. Phase I of the problem involves finding frequent item sets in the database, based on a predefined frequency threshold minsupport. Phase II of the problem involves generating the association rules from the frequent item sets found in Phase I.

Typically, the reported approaches in Phase I require multiple passes over the transaction database to determine the frequent item sets of different lengths [AS93, AS94, BMU97]. All these approaches assume that a static database is available, so that multiple scans can be made over it. With online systems, it is desirable to make decisions on the fly, processing data-streams instead of stored databases. Reported some works, which is, relates to online versions of the rule mining algorithm [CP98, CP01]. However, in these reports, reference is still made to static repositories.

In the algorithm, it has to be computed its result up to and including the first n transactions. A true online algorithm should be capable of updating the result for the (n + 1) transaction, without requiring a re-scan over the past n transactions. In this way, such an algorithm can handle transaction streams.

In fact it is true that items in an online shopping mart or a supermarket are categorized into subclasses, which in turn make up classes at a higher level, and so on. Besides the usual rules that involve individual items, learning association rules at a particular subclass or class level is also of much potential use and significance, e.g. an item-specific rule such as "Customers buying Brand A sports shoes tend to buy Brand B tee shirts" may be of less practical use than a more general rule such as "Customers buying sports shoes tend to buy tee-shirts".

3.1.1. Problem Definition : Mining Frequent Itemset

The frequent pattern-mining problem was first introduced by R. Agrawal, et al. in [AS94] as mining association rules between sets of items.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. An item set $X \subseteq I$ is a subset of items. Particularly, an item set with l items is called an l item set.

A transaction T = (tid, X) is tuple where tid is a transaction-id and X is an item set. A transaction T =(id, X) is said to contain item set Y if $Y \subseteq X$.

A transaction database TDB is a set of transactions. The support of an item set X in transaction database TDB, denoted as supTDB (X) or sup (X), is the number of transactions in TDB containing X, i.e.

$$\sup(X) = |\{(tid, Y) | ((tid, Y) \in TDB) \wedge (X \subseteq Y)\}|$$

Problem statement: Given a user-specified support threshold min sup, X is called a frequent item set or frequent pattern if sup (X) , min sup. The problem of mining frequent Item sets is to find the complete set of frequent item sets in a transaction database *TDB* with respect to a given support threshold min_ sup.

Association rules can be derived from frequent patterns. An association rule is an implication of the form $X \Rightarrow Y$, where X and Y are item sets and $X \cap Y = \phi$. The rule $X \Rightarrow Y$ has support s in a transaction database *TDB* if sup $TDB(X \cup Y) = s$. The rule $X \Rightarrow Y$ holds in the transaction database *TDB* with confidence c where $c = \frac{\sup(X \cup Y)}{\sup(X)}$.

Given a transaction database *TDB*, a support threshold minsup and a confidence threshold minconf, the problem of association rule mining is to find the complete set of association rules that have support and confidence no less than the user-specified thresholds, respectively. Association rule mining can be divided into two steps. First, frequent patterns with respect to support threshold min sup are mined. Second, association rules are generated with respect to confidence threshold minconf. As shown in many studies (e.g., [AS94]), the first step, mining frequent patterns, is significantly more costly in terms of time than the rule generation step. As we shall see later, frequent pattern mining is not only used in association rule mining. Instead, frequent pattern mining is the basis for many data mining tasks, such as sequential pattern mining and associative classification. It also has broad applications, such as basket data analysis, cross marketing, catalog design, sale campaign analysis, web log (click stream) analysis, etc.

3.1.2 Review of Apriori Algorithm

The problem of mining association rules can be decomposed into two sub problems:

Find all combinations of items whose support is greater than minimum support. Call those combinations frequent itemsets.

Use the frequent itemsets to generate the desired rules. The general idea is that if, say, ABCD and AB are frequent itemsets, then we can determine if the rule AB => CD holds by computing the ratio r = support(ABCD)/support(AB). The rule holds only if r >=minimum confidence. Note that the rule will have minimum support because ABCD is frequent.

We now present the Apriori algorithm for finding all frequent itemsets. We will use this algorithm as the basis for our presentation. Let k-itemset denote an itemset having k items. Let Lk represent the set of frequent k-itemsets, and Ck the set of candidate k-itemsets (potentially frequent itemsets). The algorithm makes multiple passes over the data. Each pass consists of two phases. First, the set of all frequent (k-1)-itemsets, L_{k-1}, found in the (k-1)th pass, is used to generate the candidate itemsets C_k.The candidate.generation procedure ensures that C_k is a superset of the set of all frequent k-itemsets.

The algorithm now scans the data. For each record, it deter-mines which of the candidates in C_k are contained in the record using a hash-tree data structure and increments their support count. At the end of the pass, C_k is examined to determine which of the candidates are frequent, yielding L_k. The algorithm terminates when L_k becomes empty. Candidate Generation Given L_k, the set of all frequent k-itemsets, the candidate generation procedure returns a superset of the set of all frequent (k+1)-itemsets. We assume that the items in an itemset are lexicographically ordered. The intuition behind this procedure is that all subsets of a frequent itemset are also frequent.

3.1.3 Apriori Algorithm

Step1
Scan the transaction database to get the support S of each 1-itemset, compare S with min_sup and get a set of frequent 1-itemsets, L1.
Step2
Use L k-1 join L k-1 to generate a set of candidate k-itemsets. And use the Apriori property(If an itemset *I* does not satisfy the minimum
support threshold, *min_sup*, the *I* is not frequent,
that is, P (*I*) < *min_sup*. • If an item A is added to the itemset *I*, then the
resulting itemset (i.e., I∪A) cannot occur more
frequently than *I*. Therefore, I ∪A is not frequent
either, that is, P (I ∪A) < *min_sup*.) to prune the unfrequent k-itemsets from this set

Step3
Scan the transaction database to get the support S of each candidate k-itemset in the final set, compare S with min_sup, and get a set of frequent k-itemsets, L k
Step4:
The candidate set = Null then goto step 2
Step5
For each frequent itemset l, generate all nonempty subsets of l
Step6
For every nonempty subset s of l,output the rule " s => (l-s)" if confidence C of the rule " s => (l-s)" (=support S of l/support S of s) ³ min_conf

Example Consider the transaction table in the figure 1. In the figure the second table gives the support count of individual items. Then the candidate item set and its support. Here in this example minimum support is as 2. After the join of two items the pruning has to be done if it required in that level. In the third level the algorithm will stop for this example.

TID	List of Items
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

1-Itemsets	Sup-count
I1	6
I2	7
I3	6
I4	2
I5	2

2-Itemsets	Sup-count
1,2	4
1,3	4
1,4	1
1,5	2
2,3	4
2,4	2
2,5	2
3,4	0
3,5	1
4,5	0

Frequent 2-Itesets	Sup-count
1,2	4
1,3	4
1,5	2
2,3	4
2,4	2
2,5	2

3-Itemsets	Sup-count
1,2,3	2
1,2,5	2

Figure 1: Example for Apriori

3.1.4 Related work in Association Rule Mining

The most well known ARM algorithm that makes use of the downward closure property is Agrawal and Srikant's Apriori algorithm [AS94]. Agrawal and Srikant used a hash tree data structure, however, Aprori can equally well be implemented using alternative structures such as set enumeration trees. Set enumeration trees impose an ordering on item and then enumerate the item sets according to this ordering. To improve this algorithm many research has been carried out, which is based on the anti monotone Apriori heuristic [AS94]: if any length k pattern is not frequent in the database, its length (k+1) super pattern can never be frequent The essential idea is to iteratively generate the set of candidate patterns of length (k+1) from the set of frequent patterns of length k (for $k \geq 1$), and check their corresponding occurrence frequencies in the database. All these algorithms are suffering mainly two problems. First: It is costly to handle a huge number of candidate sets. Second: It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching which is especially true for mining long pattern.

To solving this problem Jiawei Han et al [JIH00] proposed Frequent Pattern tree based mining method FP-growth, for mining the complete set of frequent patterns by pattern fragment growth. FP- growth method is magnitude faster for dense data but for sparse data FP-growth performs is less. Frans Coenen el al [CLA04] is proposed the Apriori- TFP algorithm perform consistently well regardless of the density of the input data set. For sparse data, Apriori-T and DIC perform better than Apriori- TFP[SFP04], while FP-growth performs less well. For dense data, FP-growth performs significantly better than Apriori- TFP, while Apriori-T and DDIC perform less well.

One of the noted work by Agarwal and Yu [AY98] considered the online generation of rules and provided a lattice based approach, called adjacency lattice. to pre-fine and prestore the primary item sets. They have analyzed the online queries and then supplied some adjacency lattice based algorithms based on the online queries received. The approach may have some following problems. If the adjacency lattice is complex and large, the processing time for constructing the lattice will be very high. It is difficult to trade off the amount of pre-stored data against the query time.

In the other work [WZ98], the authors proposed the concept of using materialized item sets for fast mining of association rules. In this approach, they divided the data base into a set of non-overlapping partitions according to some attributes. eg: education type of customers, store location, product category, and then generate the frequent item sets in each partition over the local threshold. Then the positive borders corresponding to the frequent item sets in each partition are computed. At the end all the positive borders are combined to re-mine the new frequent item sets with supports greater than the global threshold. Here in this approach is also having some problems. It will work only when the new queries with minsup larger than the presetting threshold. If users adjust the minsup below the threshold, the materialized frequent item sets fail to generate all associations. The process of frequent item sets generation needs to be performed afresh.

3.2 Negative Association Rule Mining

Typical association rules consider only items enumerated in transactions. Such rules are referred to as positive association rules. Negative association rules also consider the same items, but in addition consider negated items (i.e. absent from transactions). Negative association rules are useful in market-basket analysis to identify products that conflict with each other or products that complement each other. Mining negative association rules is a difficult task, due to the fact that there are essential differences between positive and negative association rule mining. The researchers attack two key problems in negative association rule mining: (i) how to effectively search for interesting itemsets, and (ii) how to effectively identify negative association rules of interest.

Brin et. al [BMS97] mentioned for the first time in the literature the notion of negative relationships.Their model is chi-square based. They use the statistical test to verify the independence between two variables. To determine the nature (positive or negative) of the relationship, a correlation metric was used. In [SON98] the authors present a new idea to mine strong negative rules. They combine positive frequent itemsets with domain knowledge in the form of taxonomy to mine negative associations. However, their algorithm is hard to generalize since it is domain dependant and requires a predefined taxonomy. A similar approach is described in [YBY02]. Wu et al [WCS04] derived a new algorithm for generating both positive and negative association rules. They add on top of the support-confidence framework another measure called *mininterest* for a better pruning of the frequent itemsets generated. In [THC02] the authors use only negative associations of the type $X \Rightarrow \neg Y$ to substitute items in market basket analysis

3.3 Noteworthy contributions in the field of proposed work

Online data mining of frequent item-sets over data streams for knowledge discovery is the newly emerged and rapidly growing research area and lots of challenges to be tackled cleverly to molding a proper shape and provide new dimensions in this field of work. And some of the challenges that has to be tackled are each data element over data streams should be examined at most once, the memory usage in the process of mining should be restricted as the minimum memory usage even though the new data are continuously generated from the stream and each data element in the stream should be processed as fast as possible. The different type of online queries that system can support are as follows.

a) Find all association rules above a certain level of minsupport and minconfidence.
b) At a certain level of minsupport and minconfidence, find all association rules concerned with the set of items X.
c) Find the number of association rules/itemsets in any of the cases (a),(b) above.
d) At what level of minsupport do exactly k itemset exist containing the set of items Z.

c) For a particular level minconfidence c, at what level of minsupport do exactly k-single consequent rules exists, which involve the set of items Z.

3.3.1 Online Generation of Association Rules [CP98]

This is one of the noted works by the IBM T.J. Research Center in the year 1998. We will be having a large database for do the mining technique to come up with the association rules. In this paper they are investigating the problem of online mining of association rules in the large database. For do the repeated online queries, it requires a proper preprocessing of data and while we think about the above we have to consider the available memory space also. The memory space availability is one of the big threats to be tackled effectively to come up with a useful online algorithm.

In their paper they are storing the preprocessed data in such a way that online processing can be done by applying a graph theoretic search algorithm whose complexity is proportional to the size of the output. And that leads to an online algorithm capable of finding rules with specific items in the antecedent or consequent. They have succeeded in eliminating the redundancy in some extends in this paper. More specifically contributions of their works are as follows.

a) They device a frame work for organizing the primary itemsets in such a way that online rules with very limited I/O on the pre stored data. The online time for mining the rules is independent of the size of the transaction data as well as the number of itemsets pre stored. In fact, the time required to process a query is completely dependent upon the size of the output. This feature is especially suitable for the online case.

b) They have given a technique which can quickly predict the size of the output at a given level of user specified parameters. For a given level of user-specified minsupport and minconfidence, both the number of itemsets as well as the number of rules can be predicted. A reverse query such as predicting the level of minsupport for which a particular number of itemsets exist can also be performed.

c) They have discussed the issue of efficiency in the generation of the rules. Since they included the possibility of generating rules with more than one item in the consequent, it might often be cumbersome (at least from an online perspective) to look at each of the subsets of the large itemsets as a possibility for the antecedent. A large number of possibilities can be pruned by careful order of examination. It is also possible to efficiently generate only rules with exactly one item in the consequent. Such rules are called single-consequent rules.

d) They have discussed the issue of generating rules with specific of items in them. The items might occur in the antecedent or consequent.

e) They have discussed the issue of redundancy in the rules generated from large itemsets. And also discussed the level to which essential (nonredundant) rules may often get buried in hordes of redundant rules. Compactness of representation to an online user is a very useful feature. This segment of the paper has both theoretical and practical significance.

f) They presented an algorithm for finding the primary itemsets which automatically decides which itemsets to prestore depending upon available memory capacity. The value of the primary threshold at which the best fit to this maximum number of itemsets might be found is not known in advance. One might perform a binary search on the support value in order to find the value of the primary threshold. They have proposed techniques for improving the efficiency beyond simply performing a simple binary search.

3.3.2 Online Association Rule Mining [HI99]

This is another noted work by the International Computer Science Institute, Berkely in the year 1999. In this they have proposed an algorithm to compute large itemsets online. It needs at most two scans of the transaction sequence. During the first scan the user is having the freedom to change the support threshold. Here in this work they are continuously displaying the resulting association rules along with an interval on the rule's support and confidence. During the second scan they are they are determine the precise support for each large itemset and prune all small itemsets using a new forward-pruning technique.

In this paper they have made their effort to overcome the problems in the online by their own way and it seems good and they succeeded in that. They have considered an algorithm to be online if it gives a continuous feedback, it is user controlled during processing and it yields and deterministic and accurate result. Here the algorithm named them as Continuous Association Rule Mining Algorithm (CARMA). There are two phase for this algorithm and in the first phase, it continuously constructs a lattice of all potentially large itemsets(large with respect to the scanned part of the transaction sequence). For each set in the lattice CARMA provides a deterministic lower and upper bound for its support. In this phase the user is free to adjust the support and confidence thresholds at any time. Adjusting the support threshold may result in an increased threshold for which the algorithm guarantees to include all large itemsets in the lattice. If satisfied with the rules and bounds produced so far, the user can stop the rule mining early.

During the second scan, the algorithm determines the precise support of each set in the lattice and continuously removes all small itemsets.

3.3.3 A New Approach to Online Generation of Association Rules [CP01]

This is another noted work by the IBM Research center. There are different problems that may phase by the user in the online data mining. It is hard for a user to guess a priori how many rules might satisfy a given level of support and confidence. Typically one may be interested in only a few rules. This make the problem more complicated, since a user may need to run the query multiple times in order to find appropriate levels of minsupport and minconfidence in order to mine the rules. In the other way, the problem of mining association rules may require considerable manual parameter tuning by repeated queries before useful business information can be gleaned from the transaction database.

Another issue is that while mining association rules, a large percentage of the rules may be redundant. The redundant rules have to be eliminated. For example, if the rule X=>YZ is true at a given value of minsupport and confidence, then rules such as XY=>Z, XZ=> Y, X=>Y and X=>Z are redundant. They have noted that this kind of redundancy arises when we consider rules which have more than one item in the consequent. This all are the different areas they have concentrated and came up with an acceptable result. And in their work they have given the solution for the memory problem also. There are different proofs added to back support their inventions.

3.3.4 Hierarchical Online Mining for Association Rules [NJ05]

This is another recent work in the same field from the Dhirubhai Ambani Institute of Information and Communication Technology by 2005. In this paper they have explored a hierarchical approach to online data mining. And in this they have almost met the practical requirements using a simple efficient strategy with tight bounds on the computational effort required. In this paper they have used the hierarchical approach to tackle the online mining problems.

In this method they haven't fully succeeded in the problems of the rule mining through the hierarchical approach. But they have given a hint of their next work as making the generalization of the associative rule mining in the presence of hierarchical classification. In particular, to explore the possibility of discovering rules such as XY=>Z where X, Y and Z may occur in different classes or sub classes.

3.3.5 Efficient Algorithm for Hierarchical Online Mining of Association Rules [KN06]

This is the very latest work by MindTree Consulting Private Limited and Dhirubhai Ambani Institute of Information and Communication Technology. In this paper they have made two changes from the above said work.

The efficiency is achieved by utilizing two optimizations, hierarchy-aware counting and transaction reduction, which became possible in the context of hierarchical classification. They also proposed a modified algorithm for the rule generation phase which avoids the construction of an explicit adjacency lattice.

In the first change saying that, suppose there are two classes or sub classes in set of classes of interest of which one is itself a sub-class of the other. In the just previous work the pre-transaction code is executed once for each of these elements of set of classes of interest, without taking into account this hierarchical relationship between the two. But clearly if the first iteration suggests that the current transaction does not support, say PQ**, we do not need to iterate for any of its sub-class such as PQR*.This is called as hierarchy-aware counting.

Suppose A*** and B*** are two classes of interest, and let the current transaction T be {A1Q6, A2P6, B2Q6, B1Q7, A2P7, B2P7}. While T is being checked against A***, the algorithm in fact traverse through T and finds the sub-transaction T/A***= {A1Q6, A2P6, A2P7}, which may be called the projection of class A*** on T. Clearly T/A*** does not contain any items that belong to B***, because the sub-classes of A*** and B*** are disjoint. Thus they have removed T/A*** from T and pass the remaining items T1=T-T/A*** to match against B***. Thus the part of a transaction that is a projection of a class can be removed to obtain a reduced transaction to mach against disjoint classes. This is called as transaction reduction.

4. Proposed methodology during the tenure of the research work

To achieve our proposed work first need to perform more literature survey on this field. We are concentrating much on the improvement of the algorithm in a way that reducing the search for the rules by keeping the information of the previous some queries in the memory. If we look at the online queries and its pattern it can be found that most of the queries are almost related and how it can be incorporated to reduce the work load as well as the searching time. Another important factor is the space. The space usage for the processing also has to be reduced.

So the methodology which we are proposing is as below

a) The extensive survey and performance comparisons of the existing algorithms with their relative merits and demerits.
b) Propose a suitable model to incorporate the characteristic changes of ON-LINE mining in the existing algorithms by making suitable modification.
c) Initially start with Association Rule Mining algorithm for transactional databases and consider Apriori algorithm with its variants like
    i. Multilevel ARM
    ii. Multi Dimensional ARM
    iii. ARM with constraints

iv. Hierarchical Online Mining

d) To improve the algorithm in terms of time/space complexity for possible implementation.
e) Find out the result for the duplicated search from the stored result for further usage in the rule extractions.
f) Complete the mining process in optimum phases.
g) To explore existing "Classification Rule Mining" techniques for predicting mechanism in "online" scenario.
h) Possibility study of the incorporation of hierarchy-aware counting.
i) Possibility study of the transaction reduction in the whole mining process.
j) Optimize the redundancy in the rules.
k) Collect the data to be tested during the refinement of the algorithm.
l) There is no tool available for testing the algorithm. So the existing codes of the algorithm have to be tested by developing the tool for the performance evaluation.

4.1 Information about the Data Sets

Our proposed data mining approach will require benchmarking dataset for the implementation and performance evaluation. The Dataset which is more popularly used by Data Mining Researchers. The archieve popularly known as KDDCUP Dataset. is a machine learning repository by university of Califonia Irvine, USA.

The UCI machine learning repository is a collection of databases, domain theories and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The following website link is being accessed by students, educators and researchers all over the world.

http://archive.ics.uci.edu/ml/index.html.

As an indication of the impact of the archive, it has been cited over 100 times, making it one of the top 100 most cited "papers" in all of computer science. The current version of the website was designed in 2007 by Arthur Asunction and David Newman, and this project is in collaboration with Rexa.info at the university of Massachussets Amherst. Finding support is provided from the National science foundation.

5. Expected outcome of the proposed work:

The proposed research work is aimed at Data Mining approaches designed for handling the online data streams. As discussed by the existing literature [CP98] a frame work and also the appropriate algorithms are required for efficient mining of online data. Our proposed investigation is expected to yield significant outcome in the form of modified or customized data mining algorithm for online applications. The expected outcome will be as follows:

1. An efficient approach for on-line data mining.
2. Adjacency lattice complexity will be reduced.
3. A complete on-line algorithm for online Data Mining.
4. Improve the time complexity by reducing the repetition of whole mining procedure.
5. Improvement in the total performance of the algorithm.
6. Optimized mining technique for the online data.
7. Our proposed research work is expected to yield some innovative and efficient techniques to On-line Data Mining.

6. Bibliography

[AS93] R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proceedings of the ACM SIG-MOD Conference, 1993.

[AS94] R. Agrawal and R. Srikant. Fast Algorithms for mining association rules in large database. Proceedings of the International Very Large Databases Conference, pages 487-499, 1994.

[TO96] H. Toivoneo, "Sampling large databases for association rules," in proceedings of the 22" VLDB Conference Munbai, India, pp. 134-145,1996.

[BMU97] S. Brin, R. Motwani, J. D. Ullman and S. Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data. Proceedings of the ACM SIGMOD International Conference on Management of Data, 1997.

[BMS97] Brin, S., Motwani, R. and Silverstein, C., "Beyond Market Baskets: Generalizing Association Rules to Correlations," Proc. ACM SIGMOD Conf., pp. 265-276, May 1997.

[KHS98] Koperski K., J. Han, and N. Stefanovic. An Efficient Two Step Method for Classification of Spatial Data. Proc. Symposium on spatial Data Handling, Vancouver, Canada 1998.

[WZ98] M. Wojciechowski and M. Zakrzewicz. "Itemset materializing for fast mining of association rules," in Proceedings of the 2nd East European Symposium Advances in Databases and Information Systems, ADBIS98, Poznan, Poland,Lecture Notes in Computer Science 1475,Springer. 1998.

[CP98] C.C. Aggarwal, P.S. Yu, "Online Generation of Association Rules," ICDE, pp.402, 14th International Conference on Data Engineering (ICDE'98). 1998.

[SON98] Savasere, A., Omiecinski, E., Navathe, S.: Mining for strong negative associations in a large database of customer transactions. In: Proc. of ICDE. (1998) 494–502

[HI99] C. Hidber, "Online association rule mining," in Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, 1999, pp. 145-156.

[JIH00] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining Top-K Frequent Closed Patterns without Minimum Support", Proc. 2002 Int. Conf. on Data Mining (ICDM'02), Maebashi, Japan, Dec. 2002.

[CP01] Charu C.Aggarwal and Philip S. Yu., "A New Approach to Online Generation of Association Rules", IEEE Transactions on Knowledge and Data Engineering, Vol 13, No.4, July/August 2001.

[THC02] Teng, W., Hsieh, M., Chen, M.: On the mining of substitution rules for statistically dependent items. In: Proc. of ICDM. (2002) 442 449

[YBY02] Yuan, X., Buckles, B., Yuan, Z., Zhang, J.: Mining negative association rules. In: Proc. of ISCC. (2002) 623–629

[ZU03] Zekeriya Uykan. Clustering-Based Algorithms for Single-Hidden-Layer Sigmoid Perceptron. IEEE Transactions on Neural Networks, Vol. 14, No. 3, May 2003.

[CLA04] F. Coenen, P. Leng, and S. Ahmed. Data Structure for Association Rule Mining: T-Trees and P-Trees. IEEE Tran. Of Knowledge and Data Engg, Vol. 16, No. 6, 774-778 June 2004.

[SFP 04] A Tree Partitioning Method for Memory Management in Association Rule Mining Shakil Ahmed, Frans Coenen, and Paul Leng Department of Computer Science, The University of Liverpool Liverpool L69 3BX, UK{shakil,frans,phl}@csc.liv.ac.uk ,2004.

[WCS04] Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules, ACM Transactions on Information Systems, Vol. 22, No. 3, July 2004, Pages 381–405.

[NJ05] Naresh Jotwani, "Hierarchical Online Mining for Associative Rules", International Conference on Management of Data COMAD 2005b Hyderabad, India, December 20-22,2005.

[KN06] Kishore B. Kumar and Naresh Jotwani, "Efficient Algorithm for Hierarchical Online Mining of Association Rules", International Conference on Management of Data COMAD 2006, New Delhi, India December 14-16, 2006.

[RA09] Ranjana Vyas, "Eploring Some Data Mining Technique for Decission Support System Using Efficient Knowledge Management Approach" Phd Thesis Submitted to Pt. Ravi Shankar Shukla University, Raipur, C.G.

7.List of Presented/Published papers of the candidate

[PR07] Pramod S. "Model for On-Demand GPSR based on GPSR", Proceedings of International Conference, "Sensors and Related Networks" SENNET-2007,Dec 12-14, School of Electrical Sciences,Vellore Institute of Technology,Tamilnadu, in 2007, Pages 400-405.

[PR08] Pramod S., "Financial Markets Prediction Using Quantitive Analysis" Proceedings of International Conference on "Recent Trends in Computational Science" ICRTCS-2008,11-13 June 2008, ToclH institute of Engineering and Technology, Kerala.

[PVJ09] Pramod S., O.P. Vyas, "Comparative Study of Functions Used in Stock Market Prediction" proceedings of National Conference, "Colloquim on Mobile Communication and Data & Knowledge Engineering-2009", Jan 2-3 2009,Pt. Ravishankar Shukla University, Raipur,C.G..

[PVM09] Pramod S, O.P Vyas, "Association Rule Mining for User Specified Item", accepted for the Proceedings of International Conference on "Operations Research Applications in Engineering and Management" ICOREM-2009,27-29 May 2009, Anna University,Tiruchirappally, Tamilnadu.

[PVA09] Pramod S, O.P Vyas, "ASurvey: Online Association Rule Mining Algorithms", accepted for the proceedings of International Conference on "Advanced Computing" ICAC09, 6-8 Aug-2009, Bharatidasn University, Tiruchirappalli, Tamilnadu in Aug 2009.

Signature of Supervisor

(Prof. O. P. Vyas)

S.O.S. IN COMPUTER SCIENCE
Pt. Ravishankar Shukla Univ.
RAIPUR (C.G.)

Signature of Candidate

(Mr.Pramod S.)

Forwarded

Head
S.O.S in Computer Science,
Pt.Ravishankar Shukla University,
Raipur (C.G.)

HEAD,
S. O. S. IN COMPUTER SCIENCE
Pt. RAVISHANKAR SHUKLA UNIVERSITY
RAIPUR (C. G.)